Amendments to the Claims:

This listing of claims will replace all prior version, and listings, of claims in the application:

Listing of Claims:

1-18. (canceled)

19. (currently amended) A method of creating a static control graph to support detecting constraint enforcement conflicts in a software system, the software system having at least two software elements with explicit control interactions between the software elements, and the method comprising: ~~comprising the steps of;~~

creating a new disjunctive node for each node in the software system;

creating a new conjunctive node for each ~~pair if~~ pair of corresponding constraints in the software system;

creating, for each new conjunctive node that generates and output value, a new outgoing edge from the new conjunctive node to a corresponding disjunctive node; and

creating, for each new disjunctive node that generates an output value, a new outgoing edge from the new disjunctive node to a corresponding conjunctive node.

20-25. (canceled)

26. (new) An apparatus comprising:

a machine accessible medium; and

a data structure encoded in the machine accessible medium, the data structure comprising:

conjunctive nodes and disjunctive nodes that represent characteristics of a software system; and

directed edges connecting the conjunctive nodes and the disjunctive nodes, wherein:

each of the directed edges connects two nodes as an input node to said directed edge and an output node to said directed edge;

each directed edge represents a potential influence of the input node on the output node; and

the directed edges comprise two or more edges selected from the group consisting of:

an edge that is enabled by a true value from the input mode and, when enabled, produces a true value for the output node;

an edge that is enabled by a true value from the input mode and, when enabled, produces a false value for the output node;

an edge that is enabled by a false value from the input mode and, when enabled, produces a true value for the output node; and

an edge that is enabled by a false value from the input mode and, when enabled, produces a false value for the output node.

27. (new) An apparatus according to claim 26, wherein each of the disjunctive nodes represents a Boolean guard on a functional object within the software system.

28. (new) An apparatus according to claim 26, wherein each of the conjunctive nodes represents a Boolean guard on a state change within the software system.

29. (new) An apparatus according to claim 26, wherein each of the disjunctive nodes represents a node in the software system.

30. (new) An apparatus according to claim 26, wherein each of the conjunctive nodes represents a pair of constraints in the software system.

31. (new) An apparatus comprising:

a machine accessible medium; and

a software analysis tool encoded in the machine accessible medium, the software analysis tool comprising instructions to produce, as output, a graph that exposes control interactions between software elements of a software system, the graph comprising:

conjunctive nodes and disjunctive nodes that represent characteristics of the software system; and

directed edges connecting the conjunctive nodes and the disjunctive nodes, wherein:

each of the directed edges connects two nodes as an input node to said directed edge and an output node to said directed edge;

each directed edge represents a potential influence of the input node on the output node; and

the directed edges comprise two or more edges selected from the group consisting of:

an edge that is enabled by a true value from the input mode and, when enabled, produces a true value for the output node;

an edge that is enabled by a true value from the input mode and, when enabled, produces a false value for the output node;

an edge that is enabled by a false value from the input mode and, when enabled, produces a true value for the output node; and

an edge that is enabled by a false value from the input mode and, when enabled, produces a false value for the output node.

32. (new) An apparatus according to claim 31, wherein the conjunctive nodes represent Boolean guards on state changes within the software system.

33. (new) An apparatus according to claim 31, wherein the disjunctive nodes represent Boolean guards on functional objects within the software system.

34. (new) An apparatus according to claim 31, wherein the directed edges represent implication between the conjunctive nodes and the disjunctive nodes.

35. (new) A method comprising:

generating conjunctive nodes and disjunctive nodes to represent characteristics of a software system; and

generating directed edges to connect the conjunctive nodes and the disjunctive nodes, wherein:

each of the directed edges connects two nodes as an input node to said directed edge and an output node to said directed edge;

each directed edge represents a potential influence of the input node on the output node; and

the directed edges comprise two or more edges selected from the group consisting of:

an edge that is enabled by a true value from the input mode and, when enabled, produces a true value for the output node;

an edge that is enabled by a true value from the input mode and, when enabled, produces a false value for the output node;

an edge that is enabled by a false value from the input mode and, when enabled, produces a true value for the output node; and

an edge that is enabled by a false value from the input mode and, when enabled, produces a false value for the output node.

36. (new) A method according to claim 35, further comprising:

generating, as output, a graph that includes at least one of the conjunctive nodes, at least one of the disjunctive nodes, and at least one of the directed edges.

37. (new) A method according to claim 35, further comprising:

debugging the software system, based at least in part on analysis of the conjunctive nodes, disjunctive nodes, and directed edges.

38. (new) A method according to claim 35, wherein the operations of generating conjunctive nodes, disjunctive nodes, and directed edges are performed in a data processing system, the method further comprising:

identifying a constraint conflict within the software system, based at least in part on the conjunctive nodes, disjunctive nodes, and directed edges generated in the data processing system.

39. (new) A method according to claim 35, further comprising:

storing the conjunctive nodes, disjunctive nodes, and directed edges in a data processing system:

40. (new) A method according to claim 35, further comprising:

generating a static control graph to model the software system, the static control graph to include the conjunctive nodes, disjunctive nodes, and directed edges;

selecting a first disjunctive node in the static control graph having a first incident edge from a first conjunctive node and having a first outgoing edge directed to a second conjunctive node;

checking whether the first incident edge and the first outgoing edge assert and respond to a consistent value;

proposing a new conjunctive node associated with the selected disjunctive node;

checking whether the proposed new conjunctive node is redundant of any existing conjunctive nodes in the static control graph; and

if the proposed new conjunctive node is consistent and unique, adding the new conjunctive node to the static control graph;

identifying a first set of disjunctive nodes which comprises every disjunctive node that provides an input edge to the first conjunctive node;

for each disjunctive node in the first set of disjunctive nodes, creating a corresponding sensing edge input from said disjunctive node in the first set of disjunctive nodes to the new conjunctive node;

identifying a second set of disjunctive nodes which comprises every disjunctive node other than the selected first disjunctive node that provides an input edge to the second conjunctive node;

for each disjunctive node in the second set of disjunctive nodes, creating a corresponding input edge from said disjunctive node in the second set of disjunctive nodes to the new conjunctive node;

identifying a third set of disjunctive nodes which comprises every disjunctive node that responds to an output edge from the second conjunctive node;

for each identified disjunctive node in the third set of disjunctive nodes, creating a corresponding enforcing edge from the new conjunctive node to the disjunctive node in the third set of disjunctive nodes;

identifying in the modified graph any pairs of edges that assert different values on a disjunctive node; and

determining whether the source conjunctive nodes of the identified pair of edges are mutually exclusive.

41. (new) An apparatus comprising:

a machine accessible medium; and

instructions encoded in the machine accessible medium to implement a software tool that supports detection of constraint enforcement conflicts in a software system, the software tool performing operations comprising:

creating a disjunctive node for at least one mode in the software system;

creating a conjunctive node for at least one pair of corresponding constraints in the software system; and

creating, for each conjunctive node that generates an output value, an outgoing edge from the conjunctive node to a corresponding disjunctive node.

42. (new) An apparatus according to claim 41, wherein the software system includes at least two software elements with explicit control interactions between the software elements, the software tool performing further operations comprising:

identifying a constraint conflict within the software system, based at least in part on the conjunctive nodes, disjunctive nodes, and directed edges.

43. (new) A method according to claim 19, wherein:

each of the directed edges connects two nodes as an input node to said directed edge and an output node to said directed edge;

each directed edge represents a potential influence of the input node on the output node; and

the directed edges comprise two or more edges selected from the group consisting of:

an edge that is enabled by a true value from the input mode and, when enabled, produces a true value for the output node;

an edge that is enabled by a true value from the input mode and, when enabled, produces a false value for the output node;

an edge that is enabled by a false value from the input mode and, when enabled, produces a true value for the output node; and

an edge that is enabled by a false value from the input mode and, when enabled, produces a false value for the output node.